使用 Swagger 2 构建 RESTful APIs

实践课程需求

- 使用Swagger2构建RESTful APIs
- 使用Swagger-ui构建API文档页面

Swagger

Swagger 的目标是为 REST APIs 定义一个标准的、与语言无关的接口，使人和计算机在看不到源码或者看不到文档或者不能通过网络流量检测的情况下，能发现和理解各种服务的功能。当服务通过 Swagger 定义，消费者就能与远程的服务互动通过少量的实现逻辑。类似于低级编程接口，Swagger 去掉了调用服务时的很多猜测。

Swagger 是一个简单但功能强大的 API 表达工具。它具有地球上最大的 API 工具生态系统，数以千计的开发人员，使用几乎所有的现代编程语言，都在支持和使用 Swagger。使用 Swagger 生成 API，我们可以得到交互式文档，自动生成代码的 SDK 以及 API 的发现特性等。

Spring 与 Swagger2

配置

```
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.8.0</version>
</dependency>
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger-ui</artifactId>
    <version>2.8.0</version>
</dependency>
```

- 创建 SwaggerConfig 配置类

  ```
  @Configuration
  ```

```
@EnableSwagger2
public class SwaggerConfig {
}
```

- @Configuration，启动时加载此类
- @EnableSwagger2，表示此项目启用 Swagger API 文档

```
@Bean
public Docket api() {
    return new Docket(DocumentationType.SWAGGER_2)
            .apiInfo(apiInfo())
            .select()
            // 自行修改为自己的包路径
            .apis(RequestHandlerSelectors.basePackage("com.neo.xxx"))
            .paths(PathSelectors.any())
            .build();
}
```

- 此方法使用 @Bean，在启动时初始化，返回实例 Docket（Swagger API 摘要），这里需要注意的是 .apis(RequestHandlerSelectors.basePackage("com.neo.xxx")) 指定需要扫描的包路径，只有此路径下的 Controller 类才会自动生成 Swagger API 文档。

```
private ApiInfo apiInfo() {
    return new ApiInfoBuilder()
            .title("客户管理")
            .description("客户管理中心 API 1.0 操作文档")
            //服务条款网址
            .termsOfServiceUrl("http://www.ityouknow.com/")
            .version("1.0")
            .contact(new Contact("纯洁的微笑", "http://www.ityouknow.com/", "ityoukn
ow@126.com"))
            .build();
}
```

- 这块配置相对重要一些，主要配置页面展示的基本信息包括，标题、描述、版本、服务条款、联系方式等，查看 ApiInfo 类的源码还会发现支持 license 配置等。

```
public class ApiInfo {
    public static final Contact DEFAULT_CONTACT = new Contact("", "", "");
    public static final ApiInfo DEFAULT;
```

```
    private final String version;
    private final String title;
    private final String description;
    private final String termsOfServiceUrl;
    private final String license;
    private final String licenseUrl;
    private final Contact contact;
    private final List<VendorExtension> vendorExtensions;
    //...

}
```

- 以上信息皆可在此方法进行配置，也可以使用默认值。配置完成之后启动项目，在浏览器中输入网址 http://localhost:8080/swagger-ui.html，即可看到上面的配置信息，效果如下：

Swagger 常用注解

name	age

LearnShare	12
Mike	32

作用范围	API	使用位置
协议集描述	@Api	用于 Controller 类上
协议描述	@ApiOperation	用在 Controller 的方法上
非对象参数集	@ApiImplicitParams	用在 Controller 的方法上
非对象参数描述	@ApiImplicitParam	用在 @ApiImplicitParams 的方法里边
响应集	@ApiResponses	用在 Controller 的方法上
响应信息参数	@ApiResponse	用在 @ApiResponses 里边
描述返回对象的意义	@ApiModel	用在返回对象类上
对象属性	@ApiModelProperty	用在出入参数对象的字段上

- @Api 的使用

Api 作用在 Controller 类上，做为 Swagger 文档资源，该注解将一个 Controller（Class）标注为一个 Swagger 资源（API）。在默认情况下，Swagger-Core 只会扫描解析具有 @Api 注解的类，而会自动忽略其他类别资源（JAX-RS endpoints、Servlets 等）的注解。

使用示例：

```
@Api(value = "消息", description = "消息操作 API", position = 100, protocols = "http")
@RestController
@RequestMapping("/")
public class MessageController {
}
```